基于分布的中文词表示研究 *

曹学飞，李济洪，王瑞波

(山西大学 软件学院，太原 030006)

摘　要：针对基于分布的中文词表示构造过程中的参数选择问题进行了系统性的研究。选择了六种参数进行对比实验，在中文语义相似度任务上对不同参数设置下得到的中文词表示的质量进行了评估。实验结果表明，通过选择合适的参数，基于分布的词表示在中文语义相似度任务上能够得到较高的性能，而且，这种高维的词分布表示的质量甚至优于目前流行的基于神经网络（Skip-gram）或矩阵分解（GloVe）得到的低维的词表示。

关键词：分布表示；语义相似度；逐点互信息

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2017.09.0917

Study of distributional representation of Chinese words

Cao Xuefei, Li Jihong, Wang Ruibo

(*School of Software Shanxi University*, *Taiyuan* 030006, *China*)

Abstract: To solve the problem of parameters selection in the process of constructing the distributional representations of Chinese words, this paper performed a systematic study. Six kinds of parameters were selected for comparison experiments, and the quality of the distributional representations of Chinese words obtained under different parameter settings was evaluated on the Chinese semantic similarity task. The experimental results show that, by choosing appropriate parameters, the distributional representations of Chinese words can also get higher performance on the similarity task, Moreover, the quality of such high-dimensional distributional representations is even superior to low-dimensional word representations based on neural network or matrix factorization.

Key Words: distributional representation; semantic similarity; pointwise mutual information

0　引言

词作为自然语言的基本单位，也是承载语义的基本单元，将词形式化为某种符号表示，并且这种表示能够包含词汇所要表达的语义信息，是自然语言理解与处理的基础。

目前，词的表示方法主要有基于计数的方法和基于预测的方法[1]。基于计数的方法也被称为基于分布的表示方法，它的依据是词的分布假设(distributional hypothesis)：上下文相似的词，其语义也相似[2]。因此，对语料中词-上下文之间的某种关联度量建模可刻画词的含义，一般常采用共现频次作为词-上下文之间的关联度量。具体来说，利用向量空间模型(vector space model，VSM)将词映射为语义空间中的向量，向量的维度对应词的上下文，向量元素的值表示词与上下文在一定窗口内的共现频次[3,4]，词典中所有词的向量可组织成一个矩阵，即词-上下文共现矩阵。

对于特定的任务，构造合适的 VSM 非常依赖参数[5,6]，如上下文的定义、窗口类型及大小等，不同的参数设置对模型的性能有很大的影响。如文献[3]研究了模型的性能与向量维数之间的关系，实验结果显示，通过选择合适的参数，在语义相似度任务上，使用维数为 50 K 的词的表示向量得到的结果要比 1 K 维数的向量得到的结果要好[3]；文献[6]对 VSM 涉及到的一些参数(上下文窗口的类型及大小、向量维数、语料大小以及共现频次的加权方法等)进行了系统性的研究，在四种语义任务上得到如下结论：采用 PMI(pointwise mutual information)的共现频次加权方法，尽可能小的上下文窗口以及尽可能大的向量维数得到的词的表示效果最好[6]；文献[7]在不同的英文语料上研究了 7 类参数，并在语义相似度任务上做了一系列对比实验，得到了一些经验结果：a)随着向量维数的增加，实验结果趋于稳定；b)在不同的共现频次加权方法中，PMI 是较好的选择；c)语料类型及大小对结果也有很大的影响，同一类型语料，在较大的语料上实验结果更好[7]。综上所述，目前对于 VSM 中参数的选择及设置还没有共识，且上述工作都是围绕英文来开展，就本文所知，目前还没有针对基于分布的中文词表示中参数选择问题的系统性研究，因此，本文围绕下述两个问题展开研究：

基金项目：国家社会科学规划基金资助项目（16BTJ034）

作者简介：曹学飞（1981-），男，山西清徐人，博士研究生，主要研究方向为统计自然语言处理（caoxuefei@sxu.edu.cn）；李济洪（1964-），男，教授，博士，主要研究方向为统计自然语言处理；王瑞波（1985-），男，博士研究生，主要研究方向为统计自然语言处理.

a)基于分布的词表示方法是否可以在中文上应用，其性能如何（本文以中文语义相似度任务为例）？b)在英文的分布式语义模型中所用到的参数在中文上是否有一致的的表现？基于此，本文做了详细的对比实验来探讨中文词表示的参数选择。

1 构造中文词分布表示的参数

利用 VSM 可将词表示为语义空间中的向量，VSM 的优点就在于可以很容易的使用线性代数操作计算向量之间距离从而判断词语之间的相似程度。然而，VSM 的构造涉及到很多参数，参数的选择往往决定了模型最终的性能。

1.1 词-上下文共现频次的加权方法

共现频次指的是词和上下文(仅指出现在目标词周围的一些词)在一定大小的窗口内同时出现的次数，将词表示为向量，向量的每个维度对应该词的上下文，向量的每一个元素值表示词与上下文的共现频次，这样就得到了基于分布的词的表示。然而，研究表明，直接采用原始的共现频次作为词的向量表示其效果并不好[1]，可以通过对共现频次做某些数学变换从而提高其质量，如采用不同的加权方法对共现矩阵重新赋值，常见的加权方法有 PMI、t-score 和对数似然比等[6-8]，而 PMI 是目前常用的方法之一。PMI 的定义如下：

$$PMI(w,c) = log\frac{P(w,c)}{P(w)P(c)} \tag{1}$$

其中: $P(w,c)$ 表示词 w 和上下文 c 共现的概率， $P(w)$ 和 $P(c)$ 分别表示词 w 和上下文 c 在语料中出现的概率[4]。需要注意的是，$P(w,c)$ 为 0 时的 PMI 值为负无穷，为了避免这种情况，定义此时的 PMI 值为 0。

另外，本文也考虑了 PMI 的一种变体 Positive PMI (PPMI)[6,9]作为共现频次的加权方法，PPMI 的定义如下：

$$PPMI(w,c) = \max(0, PMI(w,c)) \tag{2}$$

文献[6]的实验结果显示，在英文的语义相似度任务上，使用 PPMI 作为词的共现频次要优于 PMI。

1.2 上下文窗口的类型及大小

共现频次一般是基于上下文窗口统计得到，即在词的左右各取若干个词作为上下文，如果每一个词的窗口大小都是一个固定的常数，本文把这种方法称为固定窗口的方法(constant)。文献[10]提出了一种动态的上下文窗口方法：设置一个窗口阈值参数(thr)，每次对不同的词构造上下文时，首先生成区间在[1，thr]上的一个随机数 r，然后在目标词的前后各取 r 个词作为上下文，在这种方法下，每个词的上下文个数都是随机生成的，称之为随机窗口方法(random)。

研究表明，上下文窗口大小对词的分布表示在不同任务上的性能有很大的影响，如文献[11]指出，在句法相关的任务上应采用较小的窗口，而在语义相关的任务上则推荐较大的窗口；文献[12]发现如果度量具体名词之间的相似度，应采用较小的窗口，而如果度量抽象名词之间的相似度，则应采用较大的窗口。为考查中文语义相似度任务上合适的窗口设置，实验中本文将窗口设置为 1、2、5、8、10 和 12，即在固定窗口的方法下，分别在词的左右各取以上数值的词作为上下文，而在随机窗口的方法下，以上数值则为窗口阈值参数。

1.3 上下文平滑系数(cds)

计算 PMI 值时，可以使用从语料中统计得到的频次信息来估算相应的概率值。如使用式(3)估算上下文的概率：

$$P(c) = \frac{\#(c)}{\sum_c \#(c)} \tag{3}$$

其中 $\#(c)$ 表示上下文 c 在语料中出现的次数。但直接使用式 3 也存在一个问题，即 PMI 值会“偏向”于罕见的生僻词，即对于某些出现次数非常少的词，基于式(1)得到的 PMI 值可能会非常大[5]。因而，现有的一些研究引入一个平滑系数 α 对上下文的分布做平滑处理[9,10]：

$$P_\alpha(c) = \frac{\#(c)^\alpha}{\sum_c \#(c)^\alpha} \tag{4}$$

引入平滑处理也导致了 PMI 计算上的的变化：

$$PMI_\alpha(w,c) = log\frac{P(w,c)}{P(w)P_\alpha(c)} \tag{5}$$

在实验中分别设置平滑系数为 1 和 0.75（现有的英文实验结果显示，平滑系数取 0.75 时效果最好[9,10]）。

1.4 PMI 偏移参数（neg）

一般认为负的 PMI 值可能并不能帮助提升词的分布表示的质量，因而在一些基于英文的词分布表示的方法中会直接使用 PPMI 来代替 PMI[6,9]，而且采用 PPMI 后，共现矩阵会变的稀疏，这也更易于计算[5]。文献[9]进一步对 PPMI 做了泛化：

$$SPPMI_{neg}(w,c) = \max(0, PMI(w,c) - \log neg) \tag{6}$$

其中：neg 称为偏移参数，当 neg 取 1 时，SPPMI 等价于 PPMI；当 neg 大于 1 时，可以保留共现矩阵中关联度较高的词-上下文共现词对的互信息，同时增加了矩阵的稀疏度；当 neg 小于 1 时，共现矩阵中会包含一些并不关联的共现词对的互信息，并且会降低矩阵的稀疏度[6]。同样可以将偏移参数 neg 应用到 PMI：

$$SPMI_{neg}(w,c) = PMI(w,c) - \log neg \tag{7}$$

这样，得到了 PMI 的另外两个变体 SPMI 和 SPPMI。实验中，将 neg 的值经验性的设置为 0.2、0.5、0.8、2、5 和 8。

1.5 高频词的二次抽样(sub)

在一个大的语料中，一些高频词会出现几十万甚至上百万次，然而，这些高频词提供的有用信息却很少（如“是”、“的”等），本文可以简单的将这些高频词从语料中去掉，但这会导致许多和这些高频词相邻的词的上下文窗口变大，从而使得在原始语料下并不会共现的一些词对变得共现。文献[10]提出了一种二次抽样(*sub*)的技术，在加快计算速度的同时也一定程度上提高了词表示的质量。二次抽样的方法如下：设定一个阈值 t，当扫描语料时，出现频率大于 t 的高频词将以式（8）所示的概率值被忽略掉。

$$P = 1 - \sqrt{\frac{t}{f(w)}} \tag{8}$$

其中：$f(w)$ 为词 w 在语料中出现的频率。实验中，根据所用语料的实际情况，将 t 分别设置为 10^{-3}、0.5×10^{-3} 和 10^{-4}。

1.6 向量维数

词的分布表示的每一个向量元素代表该词的一个上下文，向量维数的大小对词的分布表示有很大的影响[3,6-7]。对于中文词的分布表示，向量维数应该怎么设置？即应该选择多少个上下文来表示一个词，或是否可以选择某种类型的词作为上下文？实验中，将词按照词频由高到低排序作为词的表示维度，并且选择了不同大小的维数和不同类型的词考查词的分布表示的质量在多大程度上依赖于向量维数。

2 语料及测试数据集

本文选用北京大学标注的人民日报 (包含 1998 年和 2000 年)作为实验语料，去掉其中包含的标注信息后语料约含 2 千 9 百万个词。

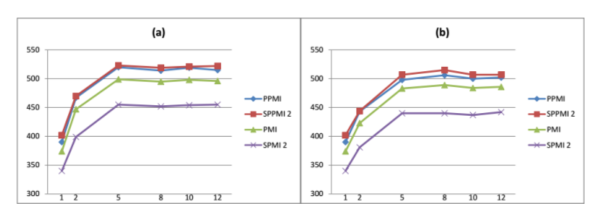
准确地解释词的语义信息需要该词所处的背景知识或上下文信息，因而很难直接去判断一个词的表示的好坏，即没有一个公认的评价指标或标准的数据集去直接评价词的分布表示的质量，只能从某些方面做间接的评估。本文选用中文语义相似度作为评测任务，该任务通过计算两个词的表示向量之间的夹角余弦来判断这两个词在语义上的相似程度[13,14]。使用 wordsim297[15]作为测试数据集，该数据集包含 297 个词对，每个词对都有人工标注的相似度得分（分值越高表示该词对的语义更相似），使用向量的夹角余弦值作为数据集中词对语义相似度的度量值，使用斯皮尔曼相关系数作为检验人工标注结果与计算结果之间相关程度的度量指标。

3 实验结果

因实验涉及参数较多，采取了一个简单的策略，即先选择一些参数分析，然后将这些参数值固定，再去研究其他的参数。实验中，首先确定窗口类型及大小，然后再考查其他参数对词的分布表示的影响。

3.1 上下文窗口的影响

首先考查在不同的上下文窗口类型及大小下，由 PMI 及其三种变体（PPMI、SPMI 和 SPPMI）得到的词的分布表示在中文语义相似度任务上的性能。SPMI 和 SPPMI 设置 *neg* 值为 2，上下文由全部出现在语料中的词构成（向量维数约为 30 万）。图 1 给出了实验结果，在固定窗口的方法下，四种加权方法基本都是在窗口为 5 时得到最优的词的分布表示，且当窗口大于 5 时，结果趋于稳定；在随机窗口方法下，PMI、PPMI 和 SPPMI 这三种加权方法的最优的窗口值为 8，当窗口大于 8 时，实验结果反而略有降低，对于 SPMI，当窗口取 12 时，实验结果虽略有提高但并不明显。图 1 中 X 轴表示窗口大小，Y 轴表示斯皮尔曼相关系数（$\rho\times1000$）。



（a）采用固定窗口方法　　（b）采用随机窗口方法

图 1　上下文窗口类型对词的分布表示的影响

图 2 为采用不同窗口大小，不同加权方法时，固定窗口方法和随机窗口方法的性能比较。结果显示，在任一种加权方法下，固定窗口方法总是优于随机窗口方法，而且即使随机窗口方法取更大的窗口值（如 12）其结果依然达不到固定窗口方法在窗口大小为 5 时的性能。因此，在后续的实验中，本文将上下文窗口的类型设定为固定窗口，大小为 5。

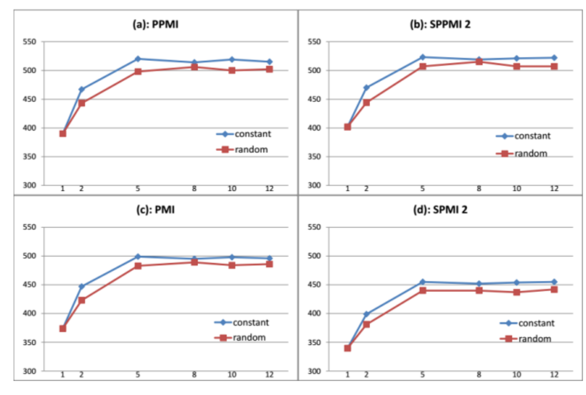


图 2　上下文窗口大小对词的分布表示的影响

（a）~（d）分别表示采用不同的共现频次的加权方法

3.2 上下文平滑系数的影响

图 3 给出了上下文平滑系数在不同的加权方法以及不同的向量维数下的实验结果。图中 X 轴表示向量维数（单位为 K），Y 轴表示斯皮尔曼相关系数（$\rho\times1000$）。本文可以很清晰地得到两个结论：

a）对上下文的分布做平滑处理（*cds* 取 0.75 时）并没有提升词的分布表示在中文语义相似度任务上的性能。

b）对于中文词的分布表示，没有必要设置更大的向量维数，实验结果显示，当维数大于 13 万时，性能并没有显著提升（图 3(b)的结果甚至略有降低）。而当维数小于 13 万并逐步增大时，性能是有明显提升的。

实验中维数为 13 万时对应的上下文在语料中都至少出现了 10 次，从图中可以看到，当本文把低频词（出现次数小于 10 次的词）从上下文中去除掉后，并不会影响词的分布表示的性能，虽然有的英文语料的实验结果认为词的分布表示的维数越大，在语义相似度任务上的性能越好[3,6]，但就中文而言，本文的结果显示，词的分布表示并不需要全部的词作为上下文，当维数达到一定数目时，在语义相似度任务上即可得到稳定的结果，再增加维数，结果没有显著性变化，反而会增加计算开销。

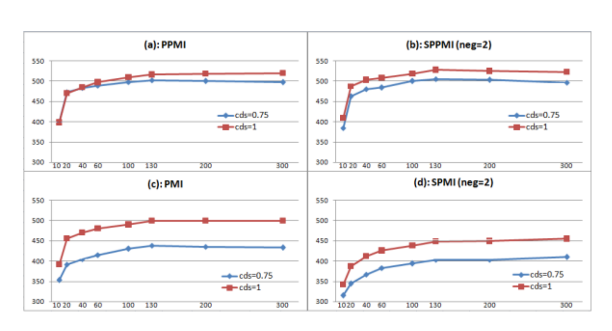


图 3　上下文平滑系数对词的分布表示的影响

（a）~(d）分别表示采用不同的共现频次的加权方法，采用固定的上下文窗口，大小为 5。

3.3　偏移参数的影响

对共现频次分别做 PMI 和 PPMI 的加权处理后再引入偏移参数 *neg*，本文发现合适的偏移参数 *neg* 可以帮助提升基于 PPMI 的分布表示的性能，但是该参数对 PMI 并没有效果。图 4(a)显示当 *neg* 值逐渐增大时，词的分布表示在语义相似度任务上的性能也在逐步提升，且当 *neg* 值取 2 时，性能达到最大；图 4(b)显示 *neg* 值无论大于 1 或小于 1，其结果均要比 *neg* 值取 1 时的结果小。本文认为，与 PMI 相比，PPMI 去掉了共现矩阵中互信息为负的词-上下文词对的共现信息，仅保留与词有一定关联的上下文信息，即对于一个词的分布表示来说，本文过滤掉了与该词并没有关联的上下文，而合适的偏移参数会进一步过滤掉关联度较低的上下文（如 4(a)中当 *neg* 取 2），但如果偏移参数过大，反而会将与词有强关联的上下文信息也舍弃掉，从而降低词的分布表示的质量（如图 4(b)中当 *neg* 取较大的 5 和 8 时，结果反而有所降低）。

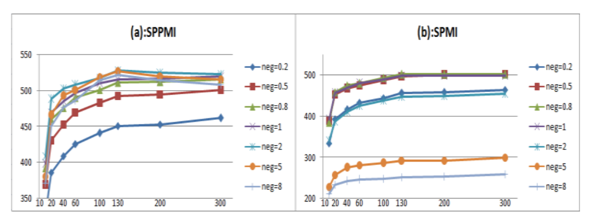


图 4　偏移参数对词的分布表示的影响

采用固定的上下文窗口，大小为 5。

3.4　高频词二次抽样的影响

二次抽样技术可以稀释语料中的高频词，效果类似于移除停用词。实验结果显示，在二次抽样对 PPMI 和 SPPMI 这两种加权方法生成的词的分布表示并没有性能上的改善，但对 PMI 和 SPMI 却有明显的帮助，如图 5(c-d)，当抽样参数 *sub* 取 10^{-4} 时，基于二次抽样技术统计得到的 PMI 和 SPMI 的分布表示在中文语义相似度的任务上性能均有提升，本文认为导致这种现象的原因是，当对 PMI 和 SPMI 的加权方法采用二次抽样技术时，相当于在一定程度上增大了窗口，使得某些词对在语料中共现的概率值增大（即式 1 的分子项增大），从而使得一些词对的 PMI（SPMI）值由负变正，这样就导致了 PMI 矩阵趋向于 PPMI 矩阵，而前文所述，PPMI 的结果要由于 PMI。但需要注意的是，PMI 和 SPMI 即使采用二次抽样，提高的性能也不足以弥补它们和 SPPMI 之间的差距(图 5)，即整体而言，四种加权方法，SPPMI 最优(见图 1-3)。

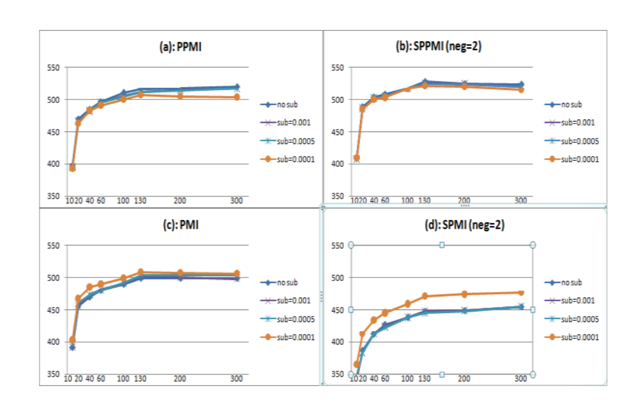


图 5　高频词二次抽样对词的分布表示的影响

采用固定的上下文窗口，大小为 5。

3.5　向量维数的选择

前面的实验结果显示，本文并不需要全部的词来作为上下文（见 3.2 节），一定程度上这是由于语料中词出现的频次服从 Zipf 定律，去掉一些低频词并不会对结果产生多大影响，正如本文实验中选择出现次数大于 10 次的词作为上下文时(约有 13 万个词)即可得到较好且稳定的词的分布表示。实际上，除了上述去掉低频词的处理，本文也采取了其他一些方法来对向量维数进行设置，如去掉停用词[6]，根据语料选取词频在一定范围内的词等[16]，但这些方法并没有带来性能的显著改善。

考虑到汉语词按照语法功能可分为实词和虚词，本文选择了实词中的名词、动词和形容词作为上下文，采用 PMI 作为共现频次的加权方法，上下文窗口为 5，不采用二次抽样技术和上下文平滑，得到如表 1 所示的实验结果。

表 1　采用实词作为上下文的实验结果

上下文类型	上下文维数	结果
语料中出现次数大于 10 次的词作为上下文	约 13 万维	0.499
实词中出现次数较多的名词、动词和形容词	约 5500 维	0.514

由实验结果可知，选用实词作为上下文，不仅减少了向量维数，从而简化了后续基于向量的一些运算操作；而且在一定程度上可以提高词语表示在中文语义相似度任务上的性能。这是因为与虚词相比，实词都是有实在意义的词，每一个实词都可以详细解说其词义，选用实词作为上下文可以更好的表示词语的意义。

3.6　与其他方法的比较

基于上述实验结果，本文可以得到如下经验结果：利用向量空间模型构造中文词的分布表示时，建议采用固定大小的窗口，窗口大小可尝试设置为 5；对共现频次做 PPMI 加权处理，并设置一个大于 1 的偏移参数(如本文中设置为 2)；而在英文上所使用的其他参数或技术本文不推荐使用，如上下文平滑系数和高频词的二次抽样技术。

另外，本文也将本文在最好的参数配置下得到的高维的分布表示(SPPMI)与目前流行的基于神经网络的 word2vec(CBOW 和 Skip-gram) [10]和基于矩阵分解的 GloVe[11]这些方法学习得到的词的低维稠密的表示在中文语义相似度任务上做了对比，表 2 中的 CBOW、Skip-gram 和 GloVe 的结果来自文献[15]，采用的训练语料与本文相同，三种方法的向量维数为 200。实验结果表明，在中文相似度任务上，基于 PPMI 的词的分布表示与上述方法是有一定的可比性的，其结果虽比 CBOW 低，但较 Skip-gram 和 GloVe 方法有明显的提高，这也印证了文献[9]的结论，该文认为，通过适当的参数设置，基于分布的词的高维表示的质量并不弱于基于神经网络或其他方法学习得到的词的低维表示(embedding)。

表 2　实验中所得的最好结果（SPPMI）与其他方法的比较

方法	CBOW	Skip-gram	GloVe	SPPMI
结果	0.556	0.517	0.431	0.528

4　结束语

本文对中文词的分布表示所涉及到的参数进行了系统性的研究，并根据实验结果给出了参数选择及设置的实用性建议。具体来说，窗口类型的选择上，固定大小的窗口优于动态窗口，窗口大小不能太小，本文推荐设置为 5（即左右各 5 个词）；对于原始的共现频次，需要做加权处理（先做 PPMI 加权，再合理的设置偏移）；向量维数即上下文个数的选择上，如果不考虑计算开销，可以简单的将所有词都设置为上下文，但更好的选择是仅用实义词来作为上下文；对于上下文平滑和二次抽样技术，本文不推荐在中文词的分布表示中使用。

本文实验基于人民日报语料，该语料虽然是人工标注，分词精度高，但与常用的英文语料相比，语料规模较小，应采用不同类型的更大的语料来对本文结果进行验证；另外，对基于共现频次得到的高维的中文词的分布表示做降维处理(如 SVD)是否可以提高其表示质量；最后，本文实验结果显示对高频词的二次抽样并没有产生在英文上的效果，是否可以通过其他方式对高频词或对某种类型的高频词进行预处理，进而得到更好的词的分布表示；以上这些都是需要本文进一步研究的。

参考文献：

[1] Baroni M, Dinu G, Kruszewski G. Don't count, predict!A systematic comparison of context-counting vs. Context predicting semantic vectors [C]// Proc of Meeting of the Association for Computational Linguistics. 2014: 238-247.

[2] Harris Z. Distributional Structure [M]. [S. l.] : Word, 1954: 146-162

[3] Milajevs D, Sadrzadeh M, Purver M. Robust Co-occurrence Quantification for Lexical Distributional Semantics [C]// Proc of ACL Student Research Workshop. 2016: 58-64.

[4] Hanks P, Hanks P. Word association norms, mutual infor-mation, and lexicography [C]// Proc of Meeting on Association for Computational Linguistics. Association for Computational Linguistics. 1989: 76-83.

[5] Turney, Peter D, Pantel, et al. From frequency to meaning: vector space models of semantics [J]. Journal of Artificial Intelligence Research, 2010, 37 (1): 141-188.

[6] Bullinaria J A, Levy J P. Extracting semantic representations from word co-occurrence statistics: A computational study [J]. Behavior Research Methods, 2007, 39 (3): 510.

[7] Kiela D, Clark S. A Systematic Study of Semantic Vector Space Model Parameters [C]// Proc of Workshop on Continuous Vector Space MODELS & Their Compositionality. 2014: 21-30.

[8] Evert S. The statistics of word cooccurrences: word pairs and collocations [D] [S. l.] : University of Stuttgart, 2004.

[9] Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings [J]. Bulletin De La Société Botanique De France, 2015, 75 (3): 552-555.

[10] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [J]. Computer Science, 2013.

[11] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[12] Hill F, Kiela D, Korhonen A. Concreteness and corpora: a theoretical and practical analysis [C]// Proc of the Workshop on Cognitive Modeling and Computational Linguistics. 2013: 75–83.

[13] 汪祥，贾焰，周斌，等. 基于中文维基百科链接结构与分类体系的语义相关度计算 [J]. 小型微型计算机系统, 2011, 32 (11): 2237-2242.

[14] 刘群，李素建. 基于《知网》的词汇语义相似度计算 [J]. 中文计算语言学, 2002 (7): 59-76.

[15] Chen X X, Xu L, Liu Z Y, et al. Joint learning of character and word embeddings [C]// Proc of International Joint Conference on Artificial Intelligence. 2015: 1236-1242.

[16] Lebret R, Collobert R. Rehabilitation of count based models for word vector representations [C]// Computational Linguistics and Intelligent Text Processing. 2015: 417-429.